Cyberbullying Detection via Multi-Model Pipelines

Evaluating Classical, Deep Learning, and Transformer-Based Approaches

by Grace Li

Abstract

This report presents a multi-model approach to detecting cyberbullying in online comments. Motivated by the severe consequences of unflagged harmful content, the study compares four pipelines (TF-IDF + neural network, sentence embeddings + BiLSTM, ensemble of classical models, and fine-tuned DistilBERT) and evaluates trade-offs in recall, precision, and deployment cost. Results show that DistilBERT achieves near-perfect performance, while classical models offer practical alternatives for constrained environments.

1. Introduction

Cyberbullying has emerged as a serious threat to mental health, particularly among adolescents. Early detection via automated text classification is crucial to flagging harmful messages and initiating intervention. However, this domain is characterized by extreme class imbalance, as most messages are non-bullying, creating a modeling challenge.

In high-stakes settings such as suicide prevention or crisis intervention, minimizing false negatives is vital. Thus, my primary objective is to maximize recall (true positive rate) while keeping false positives at a manageable level to avoid desensitizing moderators or triggering user distrust.

2. Dataset and Preprocessing

I utilized a labeled dataset from Eimear Foley's Cyberbullying repository, comprising 676 text entries (590 False, 86 True). Each line corresponds to a social media comment and a binary bullying label.

2.1 Preprocessing Pipeline

1. Cleaning: Removed special characters and normalized to lowercase.
2. Tokenization: Used NLTK's `word_tokenize`.
3. Stopword Removal: English stopwords filtered.
4. Lemmatization: Performed using WordNetLemmatizer.
5. Shuffling and Splitting: 80/20 train/test split with stratification.
6. Oversampling: SMOTE was used to balance the training set.

3. Feature Engineering

3.1 TF-IDF Vectors

TF-IDF representations with parameters `max_features=5000`,`ngram_range=(1, 2)`, `min_df=2`, and `sublinear_tf=True` were used to transform each document into sparse numerical features based on term frequency-inverse document frequency. These were used as input for both neural and classical models.

3.2 Token-Based Embeddings

To prepare data for the LSTM model, I used Keras's `Tokenizer` to convert text into integer sequences based on token frequency. The configuration was:

```
MAX_NUM_WORDS = 10000
MAX_LEN = 100
tokenizer = Tokenizer(num_words=MAX_NUM_WORDS, oov_token="<OOV>")
tokenizer.fit_on_texts(df['clean'])
sequences = tokenizer.texts_to_sequences(df['clean'])
X = pad_sequences(sequences, maxlen=MAX_LEN)
```

Only the 10,000 most frequent tokens were retained. Sequences were padded to a uniform length of 100 tokens, and the resulting dense matrix was used as input to a BiLSTM model.

3.3 BERT Tokenization

For the transformer-based model, I used HuggingFace's DistilBERT tokenizer to convert raw text into subword token IDs suitable for fine-tuning with `DistilBertForSequenceClassification`. Tokenization steps:

```
tokenizer = DistilBertTokenizerFast.from_pretrained(
                                        'distilbert-base-uncased')
def tokenize(batch):
    return tokenizer(batch['text'], truncation=True, padding=True,
                       max_length=128)
```

The tokenizer handles subword units, allowing robust representation of rare or out-of-vocabulary terms. Furthermore, truncation and padding ensured all inputs had a maximum sequence length of 128 tokens. This preprocessing was used as input for the DistilBERT model during fine-tuning.

4. Modeling Approaches

4.1 Ensemble of Classical Models

A stacked ensemble using traditional classifiers with TF-IDF features.

Base Learners:

- `LogisticRegression(class_weight='balanced', max_iter=300)`
- `RandomForestClassifier(class_weight='balanced', n_estimators=200)`
- `GaussianNB()`
- `GradientBoostingClassifier()`
- `RidgeClassifier(class_weight='balanced')`

Meta-Learner: Logistic Regression

Stacking Strategy:

- `passthrough=True` to allow meta-learner access to original features
- 3-fold cross-validation (`cv=3`)

Class Balancing: `SMOTE` applied to training data (`sampling_strategy=0.8`)

4.2 TF-IDF + Neural Network

This model uses TF-IDF features extracted from preprocessed text and feeds them into a deep fully connected neural network.

Architecture:

- Input shape: TF-IDF vector (5000-d)
- Dense(1024, ReLU) → BatchNorm → Dropout(0.5)
- Dense(512, ReLU) → BatchNorm → Dropout(0.5)
- Dense(256, ReLU) → BatchNorm → Dropout(0.3)
- Dense(128, ReLU) → BatchNorm → Dropout(0.2)
- Output: Dense(1, Sigmoid)

Loss Function: Focal loss ($\alpha = 0.25$, $\gamma = 2.0$)
Optimizer: Adam
Training: EarlyStopping and ReduceLROnPlateau callbacks used to stabilize training
Class Balancing: SMOTE with sampling_strategy=0.8

4.3 Token-Based Embeddings + BiLSTM

This model tokenizes text using Keras's Tokenizer, pads sequences to fixed length, and feeds them into a bidirectional LSTM model.

Architecture:

- Input: padded token sequence (length = 100)
- Embedding(10000 words, 128-d)
- BiLSTM(64, return_sequences=True)
- GlobalMaxPooling1D
- Dropout(0.5)
- Output: Dense(1, Sigmoid)

Loss Function: Binary Cross-Entropy
Optimizer: Adam
Input Preparation: Keras Tokenizer + SMOTE (sampling_strategy=0.8)

4.4 DistilBERT Fine-Tuning

Used HuggingFace's DistilBertForSequenceClassification on raw text inputs tokenized using DistilBERT's fast tokenizer. Fine-tuning was performed with HuggingFace's Trainer API.

Model:

```
model = DistilBertForSequenceClassification.from_pretrained(
      "distilbert-base-uncased", num_labels=2
)
```

Tokenizer: DistilBertTokenizerFast with truncation and padding (max_length=128)
Training: Performed with weighted loss and evaluation using precision/recall/F1 metrics.

5. Evaluation Strategy

Model performance was assessed using accuracy, precision, recall, and F1 Score. To improve classification balance, especially in imbalanced datasets, the optimal decision threshold was selected based on the precision-recall (PR) curve, rather than defaulting to 0.5. This allowed better control over false negatives, which is critical in a safety-sensitive task like cyberbullying detection. All metrics were computed on a held-out 20% test set.

6. Results

	Accuracy	Precision	Recall	F1 Score
TF-IDF + Ensemble	0.816	0.357	0.588	0.444
TF-IDF + NN	0.493	0.167	0.764	0.274
Embeddings + BiLSTM	0.735	0.243	0.529	0.333
BERT (after 1 epoch)	0.959	0.965	0.932	0.948
BERT (after 2 epochs)	0.987	0.995	0.983	0.987
BERT (after 3 epochs)	0.992	0.992	0.992	0.992

7. Discussion

7.1 Model Comparison

TF-IDF + Ensemble achieved the best F1 score (0.444) among non-transformer models, offering a practical balance of precision (0.357), recall (0.588), and accuracy (0.816). Its simplicity and stability make it well-suited for deployment in resource-constrained environments.

TF-IDF + Neural Network prioritized recall (0.764) at the cost of precision (0.167), resulting in a lower F1 score (0.274). It may be appropriate for applications where catching as many positive cases as possible is critical, despite higher false positives.

BiLSTM with Token Embeddings showed more balanced metrics—precision (0.243), recall (0.529), F1 (0.333)—but at increased computational cost. It provides a middle ground between traditional and transformer-based approaches, though it's more suited to research or GPU-backed systems.

BERT clearly outperformed all other models. After three epochs, it reached 0.992 across all metrics (accuracy, precision, recall, F1), indicating near-perfect generalization. Even after a single epoch, it surpassed all non-transformer baselines, demonstrating the strength of fine-tuned transformers for nuanced language understanding.

7.2 Summary

For low-cost deployment, the ensemble model offers the best trade-off. The TF-IDF neural network is useful when high recall is essential. The BiLSTM provides moderate improvement with added complexity. BERT stands out as the most powerful and reliable option, ideal for high-stakes applications demanding both precision and recall.

8. Future Work

To further improve model robustness, interpretability, and adaptability to real-world conditions, future work should focus on enhancing data diversity, enabling transparency in predictions, and ensuring long-term model relevance through continual learning and external validation.

8.1 Data Augmentation

Given the limited and imbalanced dataset, future work could explore augmentation strategies such as back-translation, synonym replacement, or generative paraphrasing to create more diverse training samples.

8.2 Model Explainability with SHAP or LIME

Incorporating explainability tools like SHAP or LIME would help illuminate which tokens or patterns drive predictions, which is crucial for gaining trust in high-stakes moderation systems.

8.3 Continuous Learning

Deploying a pipeline capable of periodically retraining or fine-tuning on newly labeled data would allow models to adapt to evolving language trends and adversarial misuse.

8.4 Real-world Feedback Integration

Human-in-the-loop systems that incorporate feedback from moderators or users could enhance recall while dynamically correcting false positives over time.

8.5 Testing on External Datasets

Current findings are based on a single labeled dataset. Benchmarking the best models against other cyberbullying or toxicity datasets (e.g., Kaggle's Jigsaw, Twitter hate speech corpora) would help validate generalizability.

9. Conclusion

This report presents a comprehensive exploration of machine learning pipelines for cyberbullying detection, emphasizing high recall and operational scalability. Beginning with traditional TF-IDF-based neural networks and progressing through BiLSTM architectures and classical ensemble models, I ultimately demonstrate that transformer-based models—specifically DistilBERT—offer a significant performance leap, achieving near-perfect classification with sufficient training.

While simpler models like ensembles provide practical, lightweight alternatives suitable for early deployment or constrained environments, BERT's superior performance highlights the power of contextualized language understanding in detecting subtle abuse. Evaluation on multiple metrics and decision threshold tuning ensures that performance reflects real-world risk trade-offs.

Future enhancements should prioritize explainability, scalability, and continuous learning, enabling these models to evolve alongside the platforms they protect. This work provides a solid foundation for responsible, high-impact NLP deployment in the fight against online harassment.